

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Česnik

**Trening z uporabo navidezne
resničnosti**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Aljaž Zrnec

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge razvijte Android mobilno aplikacijo za navidezno resničnostna očala, s pomočjo katere se lahko učimo pravilne izvedbe telovadnih vaj z lastno težo. Za implementacijo rešitve uporabite programa Unity in Blender. Vsa uporabljena orodja opišite in svojo rešitev primerjajte z že obstoječimi rešitvami na trgu.

Zahvaljujem se mentorju viš. pred. dr. Aljažu Zrnecu za uso strokovno pomoč pri izdelavi diplomske naloge. Iskrena hvala tudi svoji družini, ki me je podpirala in usmerjala, da sem vedno hodil po pravi poti. Zahvala gre tudi mojim sostanovalcem za pomoč tekom študija.

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Navidezna resničnost	3
2.1	Zgodovina razvoja	3
2.2	Najbolj uporabljene telefonske VR platforme	8
3	Uporabljena orodja in tehnologije	11
3.1	Unity	11
3.2	C#	13
3.3	MonoDevelop	14
3.4	Blender	14
3.5	Adobe Photoshop	15
3.6	VR Box	16
3.7	Samsung Galaxy Alpha	16
4	Izdelava aplikacije	19
4.1	Priprava okolja	20
4.2	Grafika	20
4.3	Animiranje	22
4.4	Programiranje	25
4.5	Izvoz programa	27

4.6	Prikaz delovanja končnega produkta	28
5	Pregled trga in izboljšave aplikacije	33
5.1	Primerjava z obstoječimi aplikacijami	33
5.2	Možne izboljšave	37
6	Sklepne ugotovitve	41
	Literatura	44

Seznam uporabljenih kratic

kratica	angleško	slovensko
VR	virtual reality	navidezna resničnost
3D	three dimensional	tri dimenzionalni
2D	two dimensional	dvo dimenzionalni
USB	universal serial bus	univerzalno serijsko vodilo
SDK	software development kit	razvojno programsko okolje

Povzetek

Naslov: Trening z uporabo navidezne resničnosti

Avtor: Blaž Česnik

Diplomsko delo zajema izdelavo 3D grafike v stereoskopskem pogledu. Aplikacija uporabniku omogoča učenje pravilne izvedbe telovadnih vaj s pomočjo mobilnih VR očal. Uporabnik lahko izbira med učenjem dveh sklopov vaj. V prvem sklopu nam moški 3D model z osmimi vajami prikaže, kako se pravilno pripraviti na trening z raztegovanjem določenih mišic, v drugem pa nam pokaže šest krepilnih vaj za trebušne mišice. Med uporabo aplikacije se lahko premikamo po prostoru in tako opazujemo izvedbo iz različnih zornih kotov. 3D model trenerja smo animirali s pomočjo programa Blender, celotno aplikacijo pa smo razvili v programu Unity s pomočjo Googlovih SDK orodji za platformo Cardboard.

Ključne besede: mobilna aplikacija, navidezna resničnost, Unity, 3D svet, Blender.

Abstract

Title: Training using virtual reality

Author: Blaž Česnik

The thesis consist of an application which is based on stereoscopic 3D gaphics. With the use of VR goggles, its function allows the user to learn the correct gym exercises. The application allows the user to choose between two sets of exercises. In the first set of exercises the 3D male model shows how to properly prepare for training by stretching certain muscles, while the second set consist of six exercises for abdominal muscles with the purpose to improve your core strength. It is possible for the user to move around the room and observe from different angles. 3D male model was animated with Blender while the application was developed by using Unity with the help of Google SDK tools for Cardboard platform.

Keywords: mobile application, virtual reality, Unity, 3D world, Blender.

Poglavje 1

Uvod

Ko povežemo navidezno resničnost s telovadbo, dobimo odličen produkt za trening raznovrstnih športov. Tu gre predvsem za pregled nad potekom treninga in izboljšanje refleksov, kar pa je temeljna lastnost dobrega atleta [1]. Če hočemo izboljšati svoje znanje pri nekem športu, je pomembno, da imamo na treningu pri vsaki vaji veliko ponovitev. V realnem svetu na to vplivajo različni dejavniki, kot so na primer vreme, športno igrišče, soigralci in trenerji. Vse to nas omejuje in tako lahko treniramo samo ob določenih pogojih. Tega problema v navideznem svetu ni, saj ima lahko športnik neskončno ponovitev, ne da bi ga karkoli omejevalo.

Zanimanje za navidezno resničnost se iz dneva v dan veča, saj nam odpira vrata v nov svet, v katerem še nikoli nismo bili. S pomočjo VR naprave lahko uporabniku predstavimo marsikatero dejavnost na bolj zanimiv način kot pa je to predstavljeno samo s pomočjo slike ali video posnetka, saj omogoča neposredno interakcijo v navideznem 3D svetu. V našem primeru nam to vzbudi dodatno zanimanje za učenje pravilne izvedbe telovadnih vaj, saj nam poleg zanimive interakcije z navideznim svetom aplikacija omogoči trening s pomočjo osebnega trenerja. Ta reagira na uporabnikovo izbiro na grafičnem vmesniku in nam tako začne prikazovati ciklične vadbne za različne mišične skupine.

V sklopu diplomske naloge smo sprogramirali aplikacijo za prikaz pravilne

izvedbe telovadnih vaj. Namen aplikacije je uporabniku omogočiti neprestan dostop do športnega trenerja. Uporabnik se lahko sprehaja po 3D prostoru in opazuje, kako 3D model človeka izvaja različne športne vaje. S pomočjo grafičnega vmesnika lahko uporabnik izbira različne sklope vaj, ustavi trenutno animacijo, nadaljuje na naslednjo vajo ali še enkrat pogleda prejšnjo vajo.

V diplomskem delu na začetku najprej na kratko opišemo zgodovino navidezne resničnosti in trenutno najbolj popularna mobilna VR očala. V tretjem poglavju opišemo uporabljena orodja in tehnologije za izdelavo diplomskega dela: igralni pogon Unity in program za 3D modeliranje in animacijo — Blender primerjamo s tehnologijami, ki služijo v namene izdelovanja 3D iger in simulacij. V osrednjem delu razložimo celoten potek razvoja aplikacije. Na koncu pa se osredotočimo na nadgradnjo in izboljšavo aplikacije, pregledamo že obstoječe programske rešitve na trgu in jih primerjamo z našo rešitvijo.

Poglavje 2

Navidezna resničnost

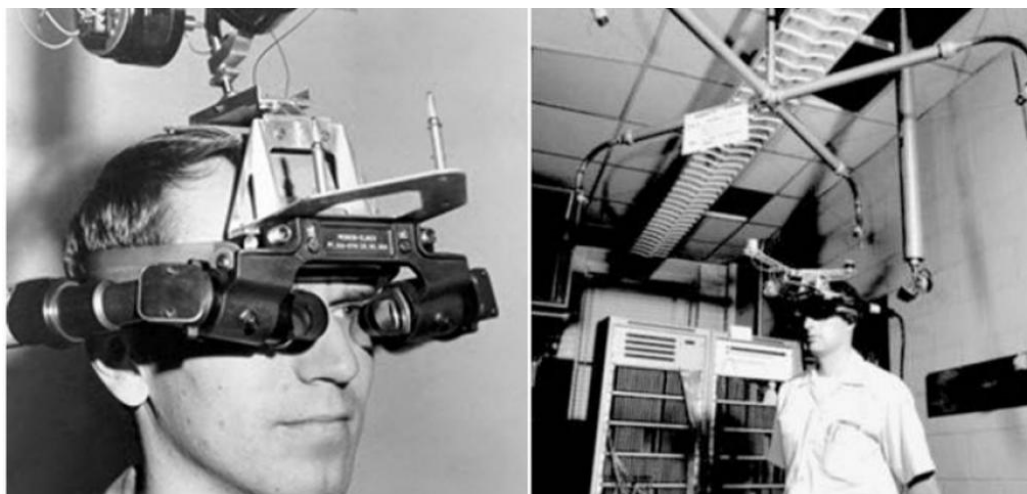
V tem poglavju je na kratko predstavljena zgodovina razvoja VR očal in nato so predstavljene še trenutno najbolj prodajane VR platforme za mobilne naprave.

2.1 Zgodovina razvoja

Ko začnemo govoriti o navidezni resničnosti, najprej pomislimo na sodobna VR očala in vse napredne aplikacije ter igre, ki jih lahko uporabljamo v današnjem času. Začetki stopnje razvoja, o kateri govorimo danes, segajo skoraj stoletje v preteklost. Razvoj navidezne resničnosti ima zato zelo pestro zgodovino, ki se začne leta 1935, ko je pisec znanstveno fantastičnih knjig Stanley G. Weinbaum v knjigi 'Pygmalion's Spectacles' [2] opisal princip delovanja očal za navidezno resničnost. Zamislil si je očala, katera bi prikazovala holografske posnetke in nam s pomočjo vonja in dotika dala občutek navidezne resničnosti.

Izdelave prvega prototipa takih očal se je lotil pionir VR tehnologije Morton Heilig. Najprej je očala leta 1950 patentiral, dvanajst let za tem pa še razvil prototip imenovan Sensorama [3]. Več-senzorska naprava je poleg možnosti prikazovanja stereoskopske simulacije 3D slike vsebovala tudi modul za zvok, vonj in veter.

Sledil je izum Ivana Sutherlanda, ki je leta 1968 s pomočjo študenta Boba Sproullija ustvaril napravo imenovano Damoklejev meč (angl. The Sword of Damocles). Ta se smatra kot prvi VR sistem, katerega si si lahko pritrdil na glavo. Stroj je bil pripet na strop, saj je tako sledil spremembam pozicije glave in na ta način prikazoval ustrezen del stereoskopskega ekrana [4]. Na sliki 2.1 je prikazan Damoklejev meč.



Slika 2.1: Prikaz Damoklejevega meča

Ime navidezna resničnost si je leta 1987 izmislil ameriški znanstvenik Jaron Lanier [5]. Zapustil je podjetje Atari in ustanovil podjetje imenovano VPL Research, ki se je ukvarjalo z razvojem in prodajo izdelkov za navidezno resničnost. Od takrat naprej so bile VR naprave dosegljive skoraj povsod [6]. V naslednjem letu je Myron Kruger izdelal prvo interaktivno VR platformo imenovano Videoplace. Odzivala se je na premikanje telesa, predvsem na gestiko rok. Videoplace je bila mešanica različnih umetno-resničnostnih sistemov, kot so: GlowFlow, Psychic Space in MetaPlay, ki so bili razviti in uporabljeni pred Videoplaceom [4].

Sledeča večja prelomnica je bila platforma Virtuality, razvita s strani podjetja Virtuality Group. VR sistem je med ljudi prišel leta 1990. Bil je namenjen igranju arkadnih iger v realnem času, z možnostjo igranja v

omrežju z več igralci.

V letu 1991 je podjetje Sega napovedalo prihod svojih Sega VR očal, ki so bila namenjena za igranje arkadnih in celo konzolnih iger. Domačo različico očal za konzolne igre so zaradi prevelikih glavobolov umaknili s prodaje. Leta 1994 je Sega izdala novo arkadno platformo imenovano Sega VR-1 [4]. Na sliki 2.2 je prikazan produkt Sega VR.



Slika 2.2: Prikaz produkta Sega VR

Na trg je leta 1995 prišel tudi prvi prenesljivi konzolni produkt podjetja Nintendo, imenovan Virtual Boy [7], naslednje leto pa produkt podjetja Forte Technologies, imenovan VFX1 (viden na sliki 2.3), ki je bil namenjen osebnim računalnikom z operacijskim sistemom DOS ali Windows 95 [4].

Po koncu 20. stoletja se je zanimanje za VR umirilo, dokler ni leta 2012 ustanovitelj podjetja Oculus VR [8], Palmer Luckey, na spletni platformi Kickstarter predstavil VR očala imenovana Oculus Rift in tako zbral okoli 2,4 milijonov dolarjev [9]. V letu 2013 je Oculus začel uporabljati zaslone podjetja Valve, saj so bili kakovostnejši in niso povzročali skoraj nič zaostanka v prikazovanju slike. Generalnega direktorja Facebooka, Marka Zuckenberg, je tehnologija, ki jo je razvilo podjetje Oculus VR navdušila, saj je v njej videl odlično orodje za razvoj VR socialnih omrežji [10]. Tako je leta 2014 za 2,3



Slika 2.3: Prva VR očala za osebne računalnike VFX1

milijarde dolarjev v denarju in delnicah kupil podjetje Oculus VR [11]. Slika 2.4 prikazuje Marka Zuckenbergga med demonstracijo platforme Oculus Rift na dogodku Oculus Connect 3.



Slika 2.4: Mark Zuckenberg med uporabo Oculus Rifta

V istem obdobju podjetji Sony in Google naznanita prihod dveh platform:

- Sonyjev Projekt Morfej (angl. Project Morpheus), VR očala za igralno konzolo PlayStation 4 [12];

-
- Googlov produkt Lepenka (angl. Cardboard), nizkocenovna platforma narejena iz kartona, leče in magneta [13].

2.2 Najbolj uporabljene telefonske VR platforme

Ker so visoko kakovostne VR naprave zelo drage in si jih ne more vsak privoščiti, je v preteklih letih nastalo kar nekaj platform namenjenim pametnim telefonom. Te naprave so v osnovi brezžične in v veliki večini stanejo manj kot 100€. V nadaljevanju so predstavljene tri najbolj uporabljene telefonske VR platforme.

2.2.1 Samsung Gear VR

Produkt je razvilo podjetje Samsung Electronics v sodelovanju z Oculus VR. Platforma je namenjena novjšim Samsungovim mobilnim produktom in ni kompatibilna z napravami drugega proizvajalca. Z očali se povežemo preko Micro-B USB konektorja in jo kalibriramo s kolesčkom lociranjem na vrhu očal. Na desni strani je sledilna ploščica za navigacijo po ekranu in štirje gumbi: gumb za nazaj, gumb za domov in dva gumba za kontroliranje glasnosti. Cena te naprave z daljincem se giblje okoli 100 € [14]. Samsung Gear VR je mogoče videti na sliki 2.5.



Slika 2.5: Očala Samsung Gear VR

2.2.2 Google Daydream View

Po Googlovih besedah je ta produkt 30 % lažji od konkurenčnih VR naprav, ki se pojavljajo na trgu. Mogoče ga je uporabiti s korekcijskimi očali, saj ima veliko prostora med lečami in očmi. Je dokaj dostopen, saj njegova cena znaša okoli 70 €, poleg same naprave pa dobimo še daljinec za interakcijo s telefonom, ki vsebuje: sledilno ploščico, funkcijski gumb, gumb za začetni zaslon, statusno lučko, gumba za spreminjanje glasnosti in priključek za polnilec [15]. Aplikacija za uporabo Daydrema je vgrajena v trenutno najnovejšo različico Androida, Nougat 7.1 [16]. Slika 2.6 prikazuje očala Google Daydream View.



Slika 2.6: Očala Google Daydream View

2.2.3 Google Cardboard

To je najcenejša različica VR očal, katero si lahko s pomočjo kartona, dveh leč in magneta, naredimo sami. Magnet pritrjen na karton deluje kot klik,

saj magnet povzroči motnjo kompasa in tako GSM telefon lahko zazna spremembo. Ker nekateri telefoni nimajo kompasa, jim je bila uporaba te funkcionalnosti onemogočena. Zato je Google izdal drugo različico očal, na katerih namesto magneta za klik uporabljajo gumb, ki se fizično dotakne ekrana [17]. S pomočjo Cardboarda je Google vzpodbudil zanimanje za VR tehnologijo, saj si ga lahko privošči vsak uporabnik pametnega telefona. Sliko očal Google Cardboard je mogoče videti na sliki 2.7.



Slika 2.7: Očala Google Cardboard

Poglavje 3

Uporabljena orodja in tehnologije

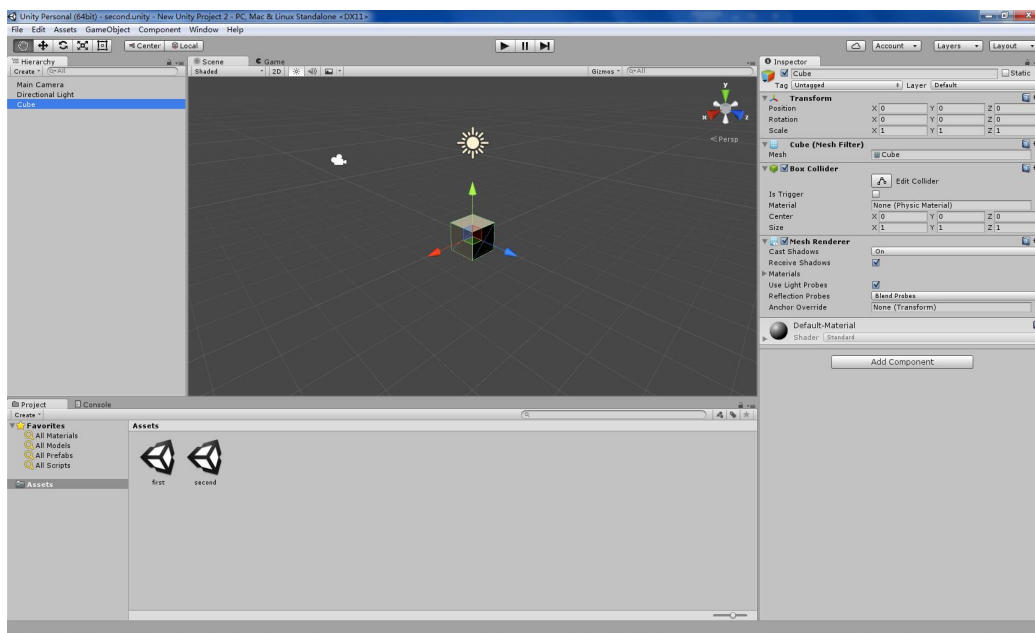
V tem poglavju so predstavljena uporabljena orodja in tehnologije, katere smo uporabili pri izdelavi aplikacije. Igralni pogon Unity in program za 3D modeliranje in animacijo — Blender primerjamo z orodji, ki so namenjena izdelovanju 3D iger in simulacij.

3.1 Unity

Igralni pogon Unity se v veliki večini uporablja za razvoj 3D in 2D iger, uporabljen pa je tudi za ustvarjanje interaktivnih aplikacij. Znan je po veliki podpori različnih platform, saj podpira preko 25 le-teh, katere so namenjene mobilnim napravam, televiziji, spletu in VR tehnologiji. Program ima velik spekter orodij, ki razvijalcem olajšajo delo in tako omogočajo hitrejši razvoj ter prihranijo veliko časa [18]. Razvijalci lahko programirajo v treh različnih skriptnih jezikih, kot so:

- C#,
- JavaScript in
- Boo.

Program je mogoče dobiti v štirih različicah, kot so Personal, Plus, Pro in Enterprise. Mi smo uporabili licenco Personal, različice 5.6.2, namenjeno posameznikom in podjetjem, ki letno zaslužijo manj kot 100.000 \$ [19]. Na sliki 3.1 je prikazano razvojno okolje programa Unity.



Slika 3.1: Razvojno okolje Unity

3.1.1 Primerjava z igralnim pogonom Unreal Engine

Na trgu je veliko orodji za izdelavo iger. Mi smo se odločili, da našo izbiro primerjamo z enim izmed najpopularnejših igralnih pogonov — Unreal Engine. Ta je marca 2015 postal brezplačen in tako se je velikemu številu začetnim razvijalcem odprlo vprašanje, kateri igralni pogon izbrati za izdelavo iger.

Če primerjamo igralna pogona med seboj, vidimo, da imata oba prednosti in slabosti. Definitivno je Unity boljša izbira za začetnike, saj ima večjo skupnost razvijalcev. Tako lahko na spletu dobimo več uporabnih nasvetov in vodičev za uporabo različnih funkcionalnosti programa, ki nam pomagajo pri izdelavi aplikacije. Na spletni trgovini AssetStore je mogoče dobiti več

izdelkov kot na trgovini orodja Unreal Engine. Pri Unityju je možnost izbire med tremi skriptnimi jeziki, v Unreal Engine pa je na voljo le jezik C++, ki ima bolj kompleksno sintakso. Če pogledamo iz stališča grafike, lahko rečemo, da ima Unreal Engine lepšo in zahtevnejšo 3D grafiko, ampak za izdelavo mobilnih aplikacij je bolj priporočljiva uporaba igralnega pogona Unity, saj so grafični elementi bolj prilagojeni zahtevam mobilnih naprav [20].

Mi smo se odločili za uporabo igralnega pogona Unity, saj imamo s tem programom že nekaj izkušenj in tako nismo imeli dodatnih težav z učenjem programa.

3.2 C#

Med tremi podprtimi programskimi jeziki, ki jih je za pisanje skript v igralnem pogonu Unity mogoče uporabiti, smo izbrali programski jezik C#. Objektno usmerjen programski jezik, ki temelji na osnovi jezika C, ima zelo preprosto sintakso in je enostaven za učenje [21]. V okviru razvoja programskega ogrodja .NET ga je razvilo podjetje Microsoft. Slika 3.2 prikazuje preprost program in berljivo sintakso programskega jezika C#.

```
1 using System;
2 namespace HelloWorld
3 {
4     class Hello
5     {
6         static void Main()
7         {
8             Console.WriteLine("Hello World");
9             Console.ReadKey();
10        }
11    }
12 }
```

Slika 3.2: Prikaz preprostega C# programa

3.3 MonoDevelop

Je integrirano razvojno okolje (angl. *integrated development environment*), ki ga je razvilo podjetje Xamarin, hčerinska družba podjetja Microsoft [22]. Pridobimo ga z namestitvijo programa Unity, kjer ga uporabimo kot program za pisanje skript. Program vsebuje različna orodja, kot so: urejevalnik kode, razhroščevalnik (angl. *debug*), prevajalnik (angl. *compiler*), samodejno dopolnjevanje (angl. *autocomplete*), itd.

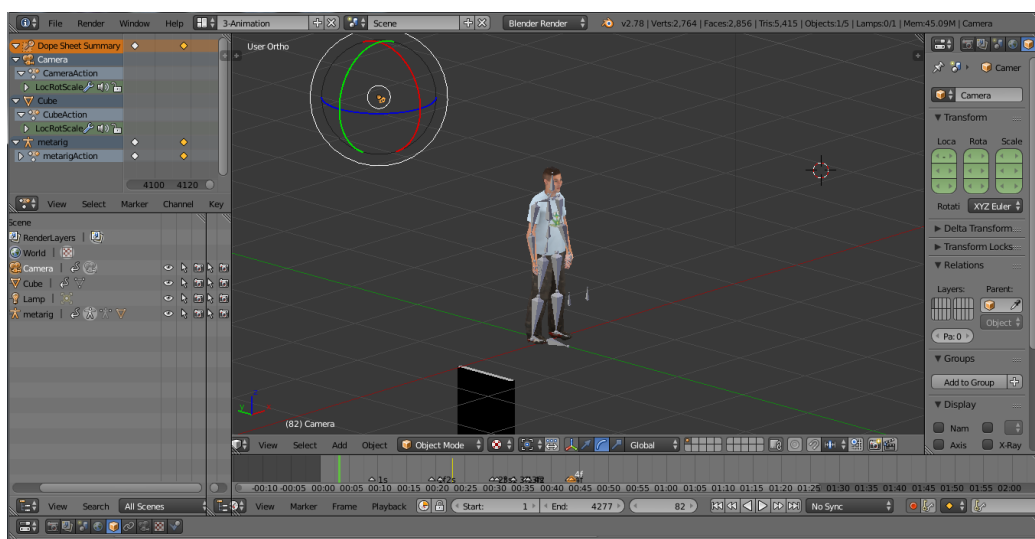
3.4 Blender

Odprtokodno programsko orodje Blender je namenjeno 3D modeliranju, ustvarjanju 3D animacij in simulacij ter izdelovanju računalniških iger. Ker je celotna uporaba programa in distribucija 3D modelov zastoj, ga uporablja veliko začetnikov in nizko proračunskih podjetij. Program podpira možnost pisanja skript v jeziku Python, s pomočjo katerih lahko ustvarimo različne vtičnike [23]. Na spletu je mogoče dobiti ogromno nasvetov in vodičev o uporabi programa, saj je skupnost uporabnikov zelo velika in povezana. Na sliki 3.3 je vidno razvojno okolje programa Blender.

Blender smo uporabili za postavitev pozicije telesa uvoženega 3D modela človeka in tako naredili potrebne animacije za našo aplikacijo. Delo smo izvajali s pomočjo programa verzije 2.78.

3.4.1 Primerjava s programom Maya

Maya je tako kot Blender program za 3D modeliranje in animacijo. Prva razlika je v ceni, saj je za uporabo programa Maya potrebno kupiti licenco, medtem ko je uporaba programa Blender brezplačna. Izbiramo lahko med mesečno, enoletno ali triletno licenco. Enoletna licenca stane okoli 2000 evrov, kar je zelo veliko in se tako začetnikom ne splača. S pomočjo programa Blender se lahko hitro naučimo osnov 3D modeliranja in animiranja, zato je primernejši za začetnike. Program Maya ima veliko dodatnih funkcionalnosti,



Slika 3.3: Razvojno okolje Blender

katere so namenjene naprednejšim uporabnikom. Če orodji primerjamo z vidika izrisovanja in senčenja, je tu boljši program Maya [24], vendar v našem primeru nismo potrebovali teh dveh funkcionalnosti.

Sami smo se odločili, da izberemo program Blender, saj smo ga v preteklosti že uporabljali.

3.5 Adobe Photoshop

Adobe Photoshop je smatran kot eden najboljših programov za obdelavo fotografij [25]. Uporabniku omogoča obrezovanje slike, spreminjanje velikosti, korekcijo barve, itd. Najbolj popularen je med fotografi in grafičnimi oblikovalci [26]. V okviru diplomskega dela smo program uporabili za popravke tekstur našega 3D modela in izdelavo ikon grafičnega vmesnika.

3.6 VR Box

Za testiranje aplikacije smo uporabili očala VR Box verzije 2.0, katera lahko uporabljamo tudi za interakcijo s Cardboard aplikacijami, saj imajo stereoskopske leče. Poleg le-teh smo v kompletu prejeli tudi Bluetooth daljinec, na katerem je igralna palica in šest funkcijskih gumbov. Slika 3.4 prikazuje očala VR Box 2.0, katera smo uporabili za testiranje aplikacije.



Slika 3.4: Očala VR Box 2.0

3.7 Samsung Galaxy Alpha

Za namene razvoja in testiranja aplikacije smo uporabili mobilni telefon Samsung Galaxy Alpha. Na telefonu smo imeli nameščen operacijski sistem Android različice 5.0.2. Naprava ima vgrajene različne senzorje, kot so: žiroskop, kompas, merilnik pospeška, itd. [27]. V nadaljevanju so predstavljeni senzorji v mobilni napravi, ki so uporabni za razvoj VR aplikacij.

3.7.1 Žiroskop

Senzor žiroskop deluje kot detektor obračanja naprave in nam tako pove, kako je naprava usmerjena. Deluje na principu vrtilne količine [28]. Pri nas žiroskop igra ključno vlogo, saj preko njega ugotovimo, v kakšnem položaju je telefon.

3.7.2 Kompas

S pomočjo kompasa lahko določimo strani neba. Magnetna igla se poravna v smer zemeljskih magnetnih silnic in nam tako pokaže smer severa [29]. Če telefonu približamo magnet, zmotimo delovanje kompasa, kar telefon zazna kot motnjo, ki jo lahko tako izkoristi za implementacijo klika.

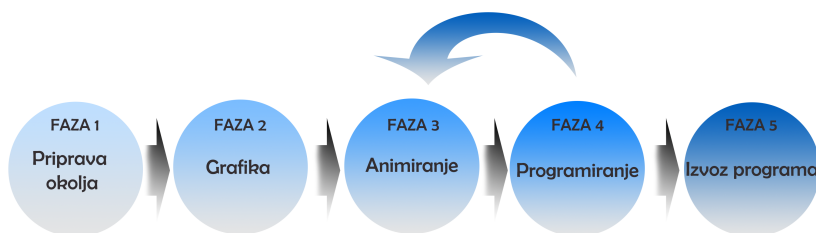
3.7.3 Merilnik pospeška

Merilnik pospeška (angl. *accelerometer*) je senzor za merjenje pospeška in gravitacije glede na premikanje ali obračanje. Le-ta zaznava tudi trke in vibracije naprave [30].

Poglavje 4

Izdelava aplikacije

Razvoj aplikacije je potekal v petih fazah. Prva faza (*Priprava okolja*) je zajemala pripravo programov in orodji za razvoj. V drugi fazi (*Grafika*) smo najprej v programu Unity oblikovali okolico in nato poiskali ustrezen 3D model, katerega smo lahko uvozili v program Blender. Sledilo je animiranje (*Animiranje*) vseh razteznih in krepilnih vaj ter vnos in priprava animacij v Unityju. V četrti fazi (*Programiranje*) smo elemente s pomočjo skript logično povezali. V zadnji fazi (*Izvoz programa*) smo program izvozili in ga pripravili za namestitvev na mobilni telefon. V nadaljevanju je podrobneje opisanih vseh pet faz poteka razvoja in prikaz delovanja končnega produkta. Na sliki 4.1 je prikazano zaporedje petih faz razvoja aplikacije.



Slika 4.1: Prikaz petih faz razvoja aplikacije

4.1 Priprava okolja

Razvoj aplikacije je potekal s pomočjo programov Blender in Unity. Programa smo uporabljali izmenično, zato smo najprej ustvarili projekt v Blenderju, nato smo na Googlovi spletni strani poiskali SDK orodje *Google VR for Unity* in ga prenesli na računalnik. Paket smo uvozili v program Unity in tako pripravili okolje za razvoj Android aplikacije. To SDK orodje omogoča zaznavo spreminjanja rotacije glave, klika ali uporabe kontrolne palice, avtomatično korekcijo žiroskopa, itd. [31].

V nadaljevanju diplomskega dela so omenjeni trije Blenderjevi načini uporabe, ki nam omogočijo dostop do različnih orodji. V programu je na voljo več načinov uporabe, vendar smo za naše potrebe potrebovali samo tri. Uporabili smo naslednje:

1. objektni način (angl. *Object Mode*),
2. urejevalni način (angl. *Edit Mode*),
3. pozicijski način (angl. *Pose Mode*).

Pri 1. načinu upravljamo s celotnim objektom. Lahko ga premikamo, obračamo ali spreminjamo velikost. V urejevalnem načinu lahko upravljamo celoten objekt ali različne dele znotraj objekta, kot so na primer posamezne ploskve. Mi smo ga uporabili za postavitev kosti na pravilen položaj, saj smo nato kosti pritrdili na objekt in ga pripravili za animacijo. 3. način je namenjen le premikanju kosti in ob tem spreminja pozicijo celotnega objekta, na katerem se kost nahaja. Uporabljali smo ga čez celotno fazo animiranja.

4.2 Grafika

4.2.1 3D model človeka

Na spletni platformi TurboSquid, kjer je mogoče dobiti ali kupiti 3D modele, smo poiskali nizko resolucijski model moškega s približno 3000 ploskvami, ki

podpira uvoz v Blender [32]. Na strani je bilo veliko modelov, ki so bili za naše potrebe izdelani v preveliki ločljivosti, saj bi to vplivalo na hitrost delovanja aplikacije. Za urejanje teksture modela smo uporabili program Adobe Photoshop.

Modelu je bilo potrebno dodati skelet človeka za možnost animacije. To smo storili v programu Blender s pomočjo dodatka Rigify [33], ki ga je bilo potrebno pred uporabo najprej namestiti v nastavitvah programa. Ta nam je zgradil preprosto skelet, katerega smo morali nato ustrezno nastaviti po sklepih našega modela. Še prej smo morali iz objektnega načina (angl. *Object Mode*) programa Blender preklopiti na urejevalni način (angl. *Edit Mode*), da smo lahko dostopali do možnosti povečave, rotacije in pozicioniranja kosti. Slika 4.2 prikazuje 3D model brez in s skeletom.



Slika 4.2: 3D model brez in s skeletom

4.2.2 Oporni zid

Pri nekaterih razteznihi vajah trener potrebuje oporo. Ker ni praktično, da se oporo konstantno vidi, saj na nekaterih pozicijah prekriva pogled končnega uporabnika, smo naredili 3D model premikajočega zidu, kateri se pojavi ob predvajanju takšne animacije. Osnovnemu modelu kocke smo povečali višino in zmanjšali širino ter tako dobili preprosti oporni zid.

4.2.3 Okolje

Za okolico smo naredili travno površino z nekaj hribčki. Orodje programa Unity omogoča hitro izdelovanje površin, tako nam ta del ni delal večjih preglavic. V programu je možen uvoz osnovnih vsebin za oblikovanje okolice, kot so premikajoča se trava, drevesa, tekstura tal, premikajoča voda, itd. Tako smo najprej oblikovali hribe okoli našega glavnega prostora, jih pobarvali z zeleno in sivo barvo ter dodali premikajočo se travo. Za osvetljevanje smo uporabili dve naravni usmerjeni svetlobi, vsaka sveti v svojo smer, saj smo tako osvetlili sprednji in zadnji del našega trenerja. Sence smo zaradi velike porabe procesorskega časa in upočasnitve aplikacije izklopili. Za izris neba smo iz Unityjeve spletne trgovine AssetStore brezplačno prenesli nebesno kockasto teksturo, v katero smo ovili naše okolje in tako dobili iluzijo pravega neba. Zraven lokacije, kjer izvaja vaje naš trener, smo za okras dodali tudi leseno konstrukcijo, ki vsebuje naprave za vadbo na prostem.

4.3 Animiranje

V tej fazi smo morali biti dodatno natančni, saj je bilo potrebno vsako okončino premakniti na določeno mesto, tako kot je pravilno za izvedbo vaje. V programu Blender smo vklopili animacijsko postavitev delovnega prostora (angl. *Animation Layout*), kar nam je odprlo različna okna, s pomočjo katerih smo lahko zanimirali naš model. Iz objektnega načina smo preklopili na pozicijski način (angl. *Pose Mode*) in začeli s postavitvijo začetnega položaja.

Najprej smo izdelali čakajočo animacijo (angl. *idle*), katera se predvaja, kadar 3D model nima izbrane vaje za prikaz. Za tem so sledile animacije za raztezanje in krepitev trebušnih mišic. Vkllopili smo avtomatično shranjevanje lokacije, rotacije in velikost kosti, saj nam tako ni bilo potrebno shranjevati vsako spremenjeno postavitev. Potek izdelave posamične animacije je izgledala tako:

1. V oknu časovnica (angl. *Timeline*) smo določili začetek animacije in postavili 3D model v začetni položaj.
2. Na časovnici smo kazalec postavili za nekaj sličic naprej in spremenili pozicijo in rotacijo določenih kosti.
3. Ko smo dosegli željeno pozicijo, kjer mišica najbolj deluje, smo preskočili nekaj sličic (angl. *frames*) in pustili trenerja v tem položaju za nekaj sekund.
4. Po izvedeni vaji smo določili konec animacije in 3D model vrnili v začetni položaj.

Model smo shranili v *fbx* formatu, saj nam je tako Blender izvozil 3D model, vse animacije in UV mapirane teksture, in ga tako pripravili za uvoz v Unity. UV mapiranje je vrsta teksturiranja, ki uporablja ploskve in robove kot regije podobe modela [34].

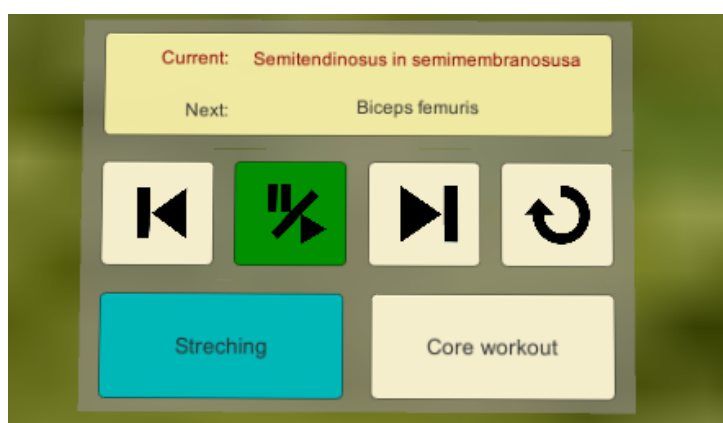
Ker smo animacije izdelali na isti časovnici, je bilo potrebno ob uvozu določiti začetke in konce vsake animacije. Za boljšo organizacijo smo posamezne animacije vaj poimenovali po številkah od 1 naprej in kot zadnji znak dodali prvo začetnico angleškega imena sklopa vaj. Tako je bila prva animacija za raztezovanje (angl. *stretching*) imenovana **1s**, naslednja **2s**, itd. Na sliki 4.3 so vidna imena animacij, številka njihove prve sličice na časovnici in številka zadnje sličice na časovnici.

Clips	Start	End
wall	86.0	126.0
metarig CubeAction	0.0	784.0
metarig metarigAction	0.0	3704.0
Idle	243.0	403.0
1s	532.0	824.0
2s	827.0	1130.0
3s	1130.0	1484.0
4s	1480.0	1924.0
5s	1924.0	2303.0
6s	2303.0	2624.0
7s	2646.0	2974.0
8s	2981.0	3688.0
hideWall	416.0	420.0

Slika 4.3: Zabeležene številke začetne in končne slike za vsako animacijo

4.3.1 Grafični vmesnik

V namen izbiranja vaj in sprehajanja med vajami smo izdelali 2D grafični vmesnik, na katerem je mogoče izbrati sklop razteznih vaj ali trebušnih krepih vaj, vstaviti trenutno pozicijo trenerja, preiti na naslednjo ali prejšnjo vajo in prav tako omogoča možnost ponovitve trenutne vaje. Slika 4.4 prikazuje interaktivni grafični vmesnik.



Slika 4.4: Prikaz grafičnega vmesnika

Na zgornjem delu vmesnika je vidno latinsko ime mišice, kateri je namenjena vaja. Za izris ikon smo si pomagali s programom Adobe Photoshop. V Unityju smo vmesnik pripeli na glavno kamero, saj tako uporablja lokacijo uporabnika in lahko ta vedno dostopa do vmesnika.

Z modro barvo je označen trenutno predvajani sklop vaj, zelena barva pa nakazuje, da je animacija v teku. Če bi bila animacija ustavljena, bi bil gumb obarvan z rdečo barvo.

4.4 Programiranje

Za pravilno prikazovanje v aplikaciji je bilo potrebno elemente logično povezati. V tem poglavju v grobem prikažemo in opišemo vsebino skript, ki smo jih napisali v okviru projekta.

4.4.1 Premikanje po prostoru

Igralni pogon Unity ima možnost nastavitve vhodnih enot, kot so miška, igralna palica ali tipkovnica, kjer definiramo različna imena sprožilcev za različne vhode. V nastavitvah smo nastavili horizontalno in vertikalno os igralne palice in gumb za izbiro. Tako smo najprej sprogramirali hojo po prostoru, katero kontroliramo z uporabo igralne palice. Kodo smo zapisali v skripto *Walk* v funkcijo `walkAround()` vidno na sliki 4.5.

```
void walkAround (){\n    Vector3 forward = Camera.main.transform.forward;\n    forward.y = 0;\n    forward = forward.normalized;\n    Vector3 right = new Vector3 (forward.z, 0, -forward.x);\n\n    float vertical = Input.GetAxis("Vertical");\n    float horizontal = Input.GetAxis("Horizontal");\n    float speed=0.09f;\n\n    transform.position += (horizontal*right + vertical*forward).normalized * speed;\n}
```

Slika 4.5: Funkcija za hojo po prostoru

S pomočjo prevzetih Unityjevih nastavitev za hojo smo se vedno premikali v eno smer in ne v smer pogleda kamere, zato je bilo to potrebno spremeniti. Preko 3D vektorja smo dobili trenutno smer kamere in vrednost ordinatne osi nastavili na 0, kar nam je preprečilo premikanje v višino. S tem smo dobili vektor za premikanje naprej in nazaj. Iz vektorja, ki je usmerjen naprej, smo izračunali vektor, ki je usmerjen v 90 stopinj od našega pogleda. To smo dosegli tako, da smo abscisni osi dodali negativni predznak. Vektor za naprej smo zmnožili z vertikalno vrednostjo in vektor usmerjen 90 stopinj od našega pogleda zmnožili s horizontalno vrednostjo. S seštevkom obeh smo izračunali smer premikanja. Dobljeni smerni vektor smo nato normalizirali in ga zmnožili z vrednostjo hitrosti premikanja. Novo izračunano lokacijo smo nastavili objektu našega pogleda. Tako smo dosegli, da je bila hoja naprej vedno v smeri pogleda kamere.

4.4.2 Predvajanje animacij

Skripto za animacije smo poimenovali *GameController* in vanjo napisali večino programske kode za interakcijo z grafičnim vmesnikom. Za vsak gumb na grafičnem vmesniku je bilo potrebno določiti, katera funkcija se izvede ob kliku nanj. Tako smo v skripti najprej javno deklarirali spremenljivke in določili reference na gumbe ter nato sprožili funkcijo za inicializacijo poslušalcev dogodkov (angl. *event listeners*), ki je prikazana na sliki 4.6.

```
void initButtonListeners(){  
  
    //Controls  
    bPlayStop.onClick.AddListener(triggerAnim);  
    bPrev.onClick.AddListener (triggerPrev);  
    bNext.onClick.AddListener (triggerNext);  
    bRepeat.onClick.AddListener (triggerRepeat);  
  
    //Exercises  
    bStretching.onClick.AddListener(stretching);  
    bCore.onClick.AddListener (core);  
}
```

Slika 4.6: Funkcija za inicializacijo poslušalcev dogodkov

4.4.3 Obračanje grafičnega vmesnika

Skripto *settingsRotation* smo namestili na naš grafični vmesnik in nastavili, da se funkcija `rotateGUI ()` sproži ob vsaki renderirani sliki. Funkcija na podlagi ordinatne osi izračuna, ali uporabnik gleda navzdol in tako onemogoči obračanje grafičnega vmesnika. To uporabniku zagotovi nemoteno izbiro po le-tem. Slika 4.7 prikazuje funkcijo za vklopjanje rotacije grafičnega vmesnika.

```
void rotateGUI (){  
    float MAngle = Mathf.Rad2Deg * Input.acceleration.y;  
    if (MAngle > 40 || MAngle < -40) {  
        float cameraYRot = camera.transform.eulerAngles.y;  
        transform.rotation = Quaternion.Euler(transform.eulerAngles.x,  
            cameraYRot, transform.eulerAngles.z);  
    }  
}
```

Slika 4.7: Funkcija za vklopjanje rotacije grafičnega vmesnika

4.5 Izvoz programa

Preden smo program dokončno izvozili iz programa Unity v končno datoteko formata *apk*, smo najprej spremenili nekaj nastavitev. V oknu za igralčeve nastavitve **Player Settings** smo za operacijski sistem Android dodali ikono naše aplikacije, katero smo prej oblikovali v Adobe Photoshopu. Isto ikono smo dodali tudi kot začetni zaslon, ampak ker uporabljamo neplačljivo verzijo programa, se nam ob vsakem zagonu aplikacije prikaže tudi ikona programa Unity. Trajanje prikaza smo omejili na 5 sekund.

V nastavitvah projekta **Project Settings** je mogoče nastaviti kakovost grafike. Izbirali smo lahko med:

1. najhitrejšo (angl. *Fastest*),
2. hitro (angl. *Fast*),
3. enostavno (angl. *Simple*),

4. dobro (angl. *Good*),
5. prelepo (angl. *Beautiful*) in
6. fantastično (angl. *Fantastic*).

Odločili smo se, da izberemo dobro (angl. *Good*) kvaliteto, saj je potem velika verjetnost, da aplikacija nima zaostanka v prikazovanju slike, poleg tega pa ima še vseeno kakovostno grafiko. Po prilagoditvi nastavitev smo v oknu za grajenje aplikacije, **Build Settings**, dokončno zgradili aplikacijo in jo pripravili na zagon. Format izvožene datoteke je *apk*, kar je standardni format za distribucijo in namestitvev Android aplikacij.

Datoteko se preko upravitelja datotek prenese na telefon in namesti na sistem. Tako je aplikacija pripravljena za uporabo.

4.6 Prikaz delovanja končnega produkta

4.6.1 Namestitev aplikacije na mobilno napravo

Pred zagonom programa, katerega smo poimenovali FITDayz, je potrebno najprej namestiti Googleovo aplikacijo Cardboard. Ta korak je potreben, ker Unity SDK orodje — GoogleVR dostopa do aplikacije in uporablja njene funkcionalnosti. Programa ni mogoče namestiti, če telefon nima žiroskopskega senzorja.

4.6.2 Uporaba aplikacije

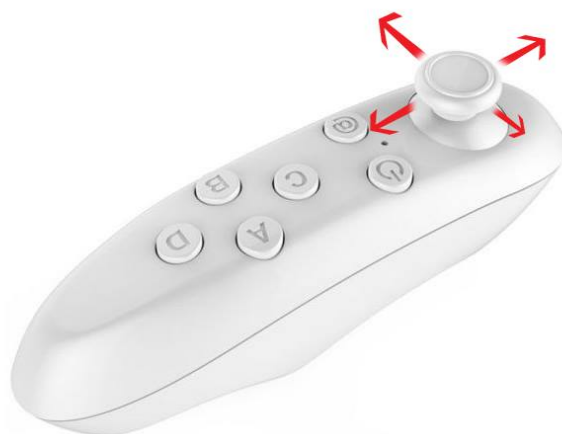
Ob zagonu aplikacije se prikaže travnata okolica z našim trenerjem, ki je v čakajočem položaju. Le-ta čaka, dokler v grafičnem vmesniku ne izberemo sklopa vaj in mu tako naročimo, kateri cikel vaj naj začne izvajati. Na sliki 4.8 je prikazan trener v mirovanju in ob izvajanju raztezne vaje.

Po grafičnem vmesniku se pomikamo s pomočjo igralne palice. Ko pridemo do zelenega gumba, ga izberemo s klikom na sprednji spodnji gumb na



Slika 4.8: Prikaz trenerja v mirovanju in ob izvajanju raztezne vaje

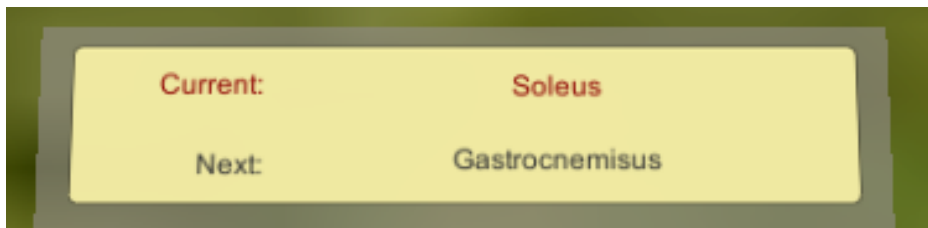
daljincu. Ob izbiri sklopa bo trener začel prikazovati prvo vajo in jo ponavljal, dokler ne izberemo ukaza za naslednjo vajo. To omogoči uporabniku, da si lahko potek vaje ogleduje več časa. Trenutno trenerjevo pozicijo lahko tudi ustavimo in bolj podrobno analiziramo položaj telesa, za ponovno predvajanje vaje pa uporabimo gumb, na katerem je narisana krožna puščica. Slika 4.9 prikazuje daljinec, katerega smo uporabili za testiranje aplikacije.



Slika 4.9: Prikaz uporabljenega daljinca

Ob gledanju vaj se lahko nemoteno sprehajamo po travniku. Ta funkcionalnost nam omogoča večji pregled nad trenerjem, saj ga lahko opazujemo iz različnih zornih kotov. Pozicijo spreminjamo s premikanjem igralne palice na našem kontrolerju.

Izbirati je mogoče med dvema sklopoma vaj: raztezne vaje (angl. *stretching*) in vaje za trebušne mišice (angl. *core*). Če izberemo raztezne vaje, začne trener izvajati 8 osnovnih vaj, katere raztegnejo mišice celega telesa. Druga izbira pa vsebuje 6 vaj za krepitev celotnih trebušnih mišic. Ob vsaki menjavi vaje se nam na vmesniku posodobi latinsko ime mišice, kateri je vaja namenjena, in ime mišice, katero bomo uporabljali v naslednji vaji. Slika 4.10 prikazuje prikaz latinskih imen mišic na grafičnem vmesniku.



Slika 4.10: Prikaz latinskega imena mišic

4.6.3 Problematika potovalne bolezni

Razvijalci VR platform poskušajo razrešiti problem potovalne slabosti (angl. *motion sickness*), kar nekaterim uporabnikom ob uporabi tehnologije povzroča slabost ali celo glavobol. Potovalna slabost oziroma morska bolezen nastane zaradi nasprotujočih si podatkov, ki jih čutila pošiljajo možganom [35]. V našem primeru ni tako velike nevarnosti, da se to zgodi, vendar možnost za pojav le-te pri nekaterih uporabnikih vseeno obstaja. Največ takih težav lahko nastane pri igranju dirkaških ali letalskih simulatorjev [36]. Tej težavi se je mogoče izogniti oziroma jo olajšati s pomočjo naslednjih postopkov:

- Če pride do takih simptomov, je priporočljivo, da imamo med uporabo aplikacije prekinitve in tako omejimo čas uporabe na nekaj minut. Čas uporabe postopoma povečujemo in telesu pomagamo premagati občutek nelagodnosti.
- Ena od metod je, da zaužijemo ingver eno ali dve uri pred uporabo VR očal. Ta nam zmanjša slabost, ker vsebuje veliko vitaminov [37].
- Obstajajo tudi tablete, katere je mogoče kupi v lekarni in so namenjene prav preprečevanju potovalne bolezni.
- Tudi marihuana blaži občutke slabosti, vendar je uporaba le-te pri nas nedovoljena oziroma je njena uporaba omejena na uporabo v medicinske namene.
- Če vam ni prijetno zauživati različnih zdravil, je mogoče na trgu kupiti zapestnico, ki pritiska na akupresurno točko na zapestju in tako lajša slabotnost.
- Uporaba aplikacije v sedečem položaj nam da občutek stabilnosti in zmanjša možnost pojavljanja takih simptomov.

Podjetja zato ob nakupu produkta priložijo dokumentacijo zdravja in varnosti, kjer so opisani mogoči simptomi uporabe [38].

Poglavje 5

Pregled trga in izboljšave aplikacije

V tem poglavju se osredotočimo na podobne programske rešitve, kot je naša aplikacija, te primerjamo z našim izdelkom in na koncu pogledamo, katere funkcionalnosti bi lahko dodali in tako nadgradili aplikacijo.

5.1 Primerjava z obstoječimi aplikacijami

Googlova trgovina Play ponuja veliko aplikacij namenjenih prikazu telovadnih vaj, izdelovanju telovadnega plana in nasvetov o zdravem življenju nasploh. V nadaljevanju poglavja opišemo tri aplikacije, ki imajo podobno vsebino kot naša.

5.1.1 Workout Revolution

Aplikacija Workout Revolution [39] naj bi bila kompatibilna s platformo Cardboard, a preklon iz navadnega načina na VR način ne deluje. Tako imamo možnost uporabe samo monoskopskega pogleda.

Pred začetkom vsake vaje se najprej izvede 10 sekundno odštevanje, v katerem se pripravimo na izvedbo vaje. Te funkcionalnosti v naši aplikaciji ni bilo potrebno implementirati, saj ni mišljeno, da uporabnik telovadi med

gledanjem animacij, ampak, da le opazuje trenerja, kako pravilno izvaja vaje. Slika 5.1 prikazuje aplikacijo Workout Revolution.



Slika 5.1: Prikaz Workout Revolution aplikacije

Pri našem izdelku je mogoče menjavati sklope vaj, medtem ko trener prikazuje vaje, tu pa je za vsako vajo potrebno v glavnem meniju najprej izbrati željeni sklop in nato vajo. Ob prikazovanju vaj nam ženski glas daje napotke za pravilno izvajanje vaje. Ta funkcionalnost je zelo koristna, saj se poleg vidne predstavitve sliši tudi dodatno glasovno razlago.

Ambient je zelo pomemben faktor, ki vpliva na počutje uporabnika. Okolje, v katerem se nahajamo je monotono, neprivlačno in ni primerljivo z našim. Aplikacija nam ne omogoča hoje po prostoru, kar posledično poslabša pregled vaje, saj je ne vidimo iz različnih zornih kotov. Ko smo primerjali našega moškega trenerja z žensko, nam je bil bolj primeren 3D model ženske, saj deluje kot bolj izkušena športnica in nam tako da občutek, da več ve.

Ob uporabi aplikacije smo opazili kar nekaj napak. Ena izmed teh je nepravilen prikaz določenih vaj, kar je pri nas ključnega pomena. Na splošno deluje naša aplikacija bolj privlačno in poučno.

5.1.2 MyFitVR

Ob zagonu aplikacije MyFitVR [40] se nam v stereoskopskem pogledu prikaže meni, kjer lahko izbiramo med različnimi kategorijami, kot so igre, vaje plesnih korakov, VR kolesarjenje itd. Mi smo se osredotočili na stereoskopsko predvajanje 360-stopinjskih posnetkov. Ob gledanju ni možna nobena interakcija z aplikacijo, saj se posnetek neprekinjeno predvaja. Vidna je telovadba osebe in ne 3D modela, zato je prikaz vaje bolj realen. Na sliki 5.2 je viden 360-stopinjski posnetek aplikacije MyFitVR.

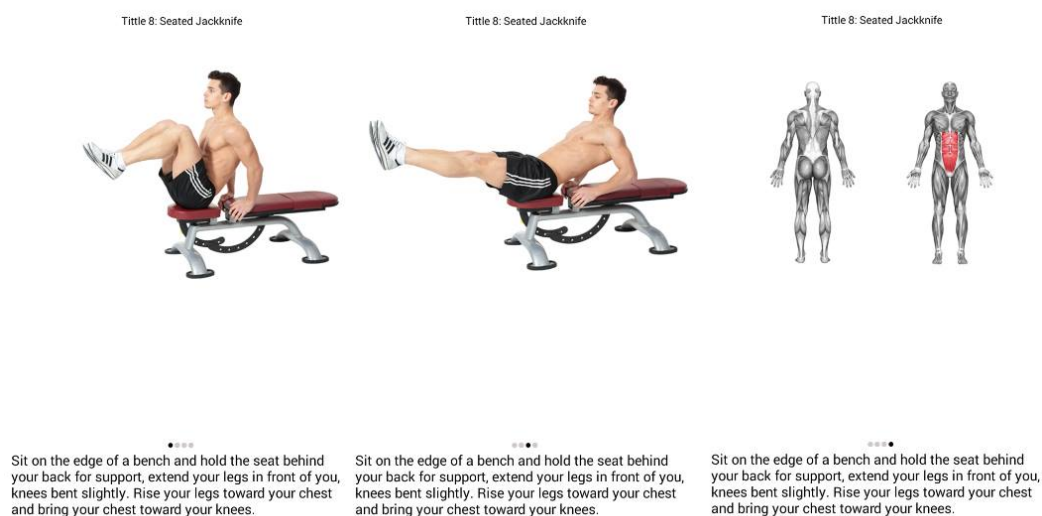


Slika 5.2: Prikaz 360-stopinjskega posnetka MyFitVR aplikacije

Tako kot prejšnja aplikacija, nam tudi ta ne omogoča premikanja po prostoru in posledično ne moramo videti športnika iz različnih kotov. To tudi ni mogoče, če uporabljamo 360-stopinjski posnetek, saj bi bilo potrebno, da kamera fizično potuje okrog moškega.

5.1.3 Gym Fitness Workouts

Aplikacija Gym Fitness Workouts [41] ne podpira platforme Cardboard, a vseeno dobro razloži izvedbo vaj za različne mišične skupine. Poleg tega ima uporabnik na voljo različne programe treningov za bodibuilding, fitnes in triatlon moči (angl. *powerlifting*). Slika 5.3 prikazuje galerijo aplikacije Gym Fitness Workouts.



Slika 5.3: Prikaz galerije Gym Fitness Workouts aplikacije

Ko izberemo mišično skupino, se nam pokaže meni z različnimi vajami. Po izbiri ene izmed vaj se nam odpre galerija s tremi ali štirimi slikami, pod njo pa je opis za pravilno izvajanje. Na slikah je prikazan moški, ki izvaja vajo, razen na zadnji sliki galerije, kjer je z rdečo barvo označena primarno fokusirana mišica.

Aplikacija je zasnovana na drugačen način kot pri nas, saj so vaje prikazane le s slikami in opisom in ne v 3D okolju. Navdušil nas je prikaz fokusirane mišice, saj tako točno vemo, katero mišico uporabljamo pri določeni vaji. Za razliko od naše aplikacije so za izvajanje večine vaj potrebne dodatne naprave ali pripomočki, kot so uteži, vadbeni trakovi itd. V oči so nam padli tudi zanimivi motivacijski citati, katere lahko berete ob uporabi programa.

5.1.4 Rezultati primerjave aplikacij

V tabeli 5.1 so vidni rezultati primerjave različnih kategorij, naše aplikacije z drugimi.

Ime aplikacije	Grafika	Prikaz vaje	Interakcija z grafičnim vmesnikom	Povprečna ocena
FITDayz	8	8	10	9
Workout Revolution	9	7	8	8
MyFitVR	6	8	7	7
Gym Fitness Workouts	6	10	10	9

Tabela 5.1: Prikaz rezultatov primerjave aplikacij

5.2 Možne izboljšave

Našo aplikacijo je možno nadgraditi. V nadaljevanju je predstavljenih nekaj možnih izboljšav.

5.2.1 Spreminjanje okolja in trenerja

Ljudje imamo različne okuse za vsako stvar, v našem primeru gre za prostor, kjer se aplikacija dogaja. Nekateri ljudje radi telovadijo na prostem, drugi v telovadnicah, nekateri pa celo ob morju. V programu bi dodali možnost za izbiro okolja in tako omogočili uporabniku spremembo ambienta po lastni želji. Mogoče bi bilo spremeniti moškega trenerja v žensko, ali pa ga samo preobleči.

5.2.2 Plačilo za dodatno vsebino

Aplikacija bi lahko vsebovala navidezni denar, katerega bi kupili s pomočjo platforme PayPal. Z denarjem bi lahko odklenili v osnovi nedostopno vsebino, kot so cikli vaj, različna okolja ali pa celo trenerje in njihova oblačila.

5.2.3 Dodajanje vaj

Modul za dodajanje vaj bi bil zelo koristna zadeva, saj bi lahko upravljalec aplikacije neprestano dodajal novo vsebino. V našem primeru bi bilo potrebno animacije izdelati v programu Blender in zraven dodati še tekstovno datoteko, kjer so zapisane vse vaje cikla in številka začetne in končne sličice vaje. Tako bi s pomočjo skripte prebrali tekstovno datoteko in prek nje zgenerirali animacije.

5.2.4 Pripovedovalec pravilne izvedbe vaje

Uporaba avdio razlage bi uporabniku dodatno pojasnila, kako se vajo pravilno izvaja. Idejo smo dobili, ko smo v poglavju 5.1.1. primerjali našo aplikacijo z *Workout Revolution*.

5.2.5 Prikaz kotov gibanja

Ko gre za izvedbo specifične vaje, je pomembno, za koliko se neka okončina premakne, nagne itd. V animaciji trenerja bi lahko uporabili posebne črte, ki bi prikazovale pomembne kote pri gibanju. To bi uporabniku aplikacije dalo dodaten napotek za pravilen gib.

5.2.6 Uporaba senzorjev gibanja

Pri animiranju naših animacij smo porabili veliko časa za to, da smo vsak sklep posebej postavili na pravilno lokacijo. To bi lahko še hitreje naredili z uporabo obleke za zajemanje gibanja v prostoru (angl. *motion capture*).

Cene teh oblek in vsa potrebna oprema se giblje od 1000 € naprej. Na sliki 5.4 so prikazana dekleta, oblečena v obleko za zajemanje gibanja.



Slika 5.4: Dekleta oblečena v obleko za zajemanje gibanja

Poglavje 6

Sklepne ugotovitve

Google je s produktom Cardboard omogočil širši množici uporabnikov mobilnih telefonov dostop do VR. To je povečalo osveščenost na področju te tehnologije in tako je iz dneva v dan zanjo vse več zanimanja. Kot razvijalcem nam to odpre večje možnosti uspeha, saj je takšnih aplikacij na trgu zelo malo. Če primerjamo trg aplikacij s podobnimi funkcionalnostimi kot jih ima naša, jih skoraj ni. Je pa veliko takih, ki služijo istemu namenu in za predstavitev uporabljajo 360-stopinjski pogled video posnetkov ali pa samo slike izvedbe vaj.

V okviru diplomske naloge smo razvili aplikacijo namenjeno Google Cardboard platformi z naslovom FITDayz. Namen te je uporabnika naučiti pravilne izvedbe telovadnih vaj in ga tako spodbuditi k aktivni telovadbi. Za uporabo orodji, ki smo jih za razvoj uporabili, je potrebno programersko znanje in občutek za 3D animacijo. Če so animacije narejene površno, je v našem primeru aplikacija neuporabna, zato je bila natančnost animacij in prikazovanja le-teh zelo pomembna.

V začetku smo namesto uporabe daljinca in igralne palice uporabljali le žiroskopski senzor. S pomočjo zaznave nagiba glave smo nato izbrali smer obračanja in tako sprožili uporabnikovo premikanje v krožni smeri okoli našega trenerja. Ker smo imeli na voljo Bluetooth kontroler, smo se odločili, da to nadgradimo in izkoristimo funkcionalnost kontrolerja. To nam

je omogočilo premikanje po celotnem navideznem prostoru in ne samo v krožni smeri.

Lahko rečemo, da je razvoj VR tehnologij že na visokem nivoju, vendar je še vedno potrebnih veliko izboljšav. Ena od teh je grafika, ki bi naši aplikaciji dala dodatno vrednost, saj je v našem interesu, da navidezno resničnost čim bolj približamo realnemu svetu in tako izboljšamo uporabnikovo izkušnjo.

Literatura

- [1] STRIVR. Sports - Take Training To The Next Level[Online]. Dosegljivo: <http://strivrlabs.com/sports/>. [Dostopano: 27. 6. 2017].
- [2] Stanley G. Weinbaum - Pygmalions Spectacles[Online]. Dosegljivo: <http://www.simonandschuster.co.uk/books/Pygmalions-Spectacles/St Stanley-Grauman-Weinbaum/9781609772307#>. [Dostopano: 3. 7. 2017].
- [3] Virtual Reality, Wikipedia[Online]. Dosegljivo: https://en.wikipedia.org/wiki/Virtual_reality. [Dostopano: 5. 7. 2017].
- [4] Proteus VR Labs LTD. Time Travel Through Virtual Reality [Online]. Dosegljivo: <https://freeflyvr.com/time-travel-through-virtual-reality/>. [Dostopano: 4. 7. 2017].
- [5] Virtual Reality Society - Who Coined the Term “Virtual Reality”?s[Online]. Dosegljivo: <https://www.vrs.org.uk/virtual-reality/who-coined-the-term.html>. [Dostopano: 4. 6. 2017].
- [6] Stanley G. Weinbaum - The virtual curmudgeon[Online]. Dosegljivo: <http://www.economist.com/node/16909935>. [Dostopano: 5. 7. 2017].
- [7] Wikia Nintendo - Virtual Boy[Online]. Dosegljivo: http://nintendo.wikia.com/wiki/Virtual_Boy. [Dostopano: 8. 7. 2017].
- [8] The Guardian. Nellie Bowles - First Oculus Rift virtual reality headsets shipped to Kickstarter backers[Online]. Dose-

- gljivo: <https://www.theguardian.com/technology/2016/mar/28/oculus-rift-virtual-reality-headsets-shipped-facebook>. [Dostopano: 8. 7. 2017].
- [9] Kickstarter - Oculus Rift: Step Into the Game[Online]. Dosegljivo: <https://www.kickstarter.com/projects/1523379957/oculus-rift-step-into-the-game?ref=discovery>. [Dostopano: 8. 7. 2017].
- [10] Facebook. Mark Zuckerberg[Online]. Dosegljivo: <https://www.facebook.com/zuck/posts/10101319050523971>. [Dostopano: 9. 7. 2017].
- [11] The Guardian. Stuart Dredge - Facebook closes its \$2bn Oculus Rift acquisition. What next? [Online]. Dosegljivo: <https://www.theguardian.com/technology/2014/jul/22/facebook-oculus-rift-acquisition-virtual-reality>. [Dostopano: 9. 7. 2017].
- [12] VentureBeat, Project Morpheus: Sony announces its virtual-reality headset for PlayStation 4[Online]. Dosegljivo: <https://venturebeat.com/2014/03/18/project-morpheus-sony-announces-its-virtual-reality-headset-for-playstation-4/>. [Dostopano: 6. 7. 2017].
- [13] TIME, Jared Newman - The Weirdest Thing at Google I/O Was This Cardboard Virtual Reality Box[Online]. Dosegljivo: <http://time.com/2925768/the-weirdest-thing-at-google-io-was-this-cardboard-virtual-reality-box/>. [Dostopano: 6. 7. 2017].
- [14] Trusted Reviews. Samsung Gear VR[Online]. Dosegljivo: <http://www.trustedreviews.com/reviews/samsung-gear-vr>. [Dostopano: 12. 7. 2017].

-
- [15] WEAREABLE. Lily Prasuethsut - Google Daydream View review[Online]. Dosegljivo: <https://www.wareable.com/vr/google-daydream-view-review>. [Dostopano: 12. 7. 2017].
- [16] WIRED. Matt Burgess - Android Nougat 7.1 'coming later this month' with support for Google's Daydream VR[Online]. Dosegljivo: <http://www.wired.co.uk/article/android-nougat-7-1-beta-developers>. [Dostopano: 12. 7. 2017].
- [17] gearbrain. Maria Korolov - Review: Google Cardboard V1 vs Google Cardboard V2[Online]. Dosegljivo: <https://www.gearbrain.com/review-google-cardboard-v1-vs-v2-1622025361.html>. [Dostopano: 13. 7. 2017].
- [18] Unity Technologies. Unity Products - Features [Online]. Dosegljivo: <https://unity3d.com/unity/features/multiplatform>. [Dostopano: 10. 7. 2017].
- [19] Unity Technologies. Unity Store [Online]. Dosegljivo: <https://store.unity.com/>. [Dostopano: 10. 7. 2017].
- [20] stfalcon. Anton Bevza - Unity3D or Unreal Engine 4[Online]. Dosegljivo: <https://stfalcon.com/en/blog/post/unity3d-vs-unreal-engine-4>. [Dostopano: 31. 7. 2017].
- [21] C Sharp. Wikipedija [Online]. Dosegljivo: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)). [Dostopano: 10. 7. 2017].
- [22] MonoDevelop - Wikipedia[Online]. Dosegljivo: <https://en.wikipedia.org/wiki/MonoDevelop>. [Dostopano: 29. 7. 2017].
- [23] Blender. Wikipedija [Online]. Dosegljivo: <https://sl.wikipedia.org/wiki/Blender>. [Dostopano: 10. 7. 2017].

-
- [24] Slant. Blender vs AutoDesk Maya[Online]. Dosegljivo: https://www.slant.co/versus/5501/5506/~blender_vs_autodesk-maya. [Dostopano: 31. 7. 2017].
- [25] The Top Tens. Best Graphic Design Software Programs[Online]. Dosegljivo: <https://www.thetoptens.com/best-graphic-design-software-programs/>. [Dostopano: 15. 7. 2017].
- [26] BusinessDictionary. Photoshop [Online]. Dosegljivo: www.businessdictionary.com/definition/Photoshop.html. [Dostopano: 15. 7. 2017].
- [27] GSM Arena. Samsung Galaxy Alpha[Online]. Dosegljivo: http://www.gsmarena.com/samsung_galaxy_alpha-6573.php. [Dostopano: 23. 7. 2017].
- [28] MIROSLAV MATIJEVIĆ. Miselna igra za platformo google cardboard. 2017.
- [29] Kompas. Wikipedia [Online]. Dosegljivo: <https://sl.wikipedia.org/wiki/Kompas>. [Dostopano: 10. 8. 2017].
- [30] Aljaž Potočnik David Osredkar. Merilniki pospeška. 2011.
- [31] Google. Developers - Unity VR[Online]. Dosegljivo: <http://developers.google.com/vr/unity/>. [Dostopano: 7. 6. 2017].
- [32] TurboSquid. 3D models - Patrick [Online]. Dosegljivo: <https://www.turbosquid.com/3d-models/human-male-3d-model/504883>. [Dostopano: 4. 6. 2017].
- [33] Rigify. Wiki Blender [Online]. Dosegljivo: <https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Rigging/Rigify>. [Dostopano: 10. 7. 2017].

- [34] MIZS. Računalniško oblikovanje slovarček[Online]. Dosegljivo: http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/Strukturni_skladi/Gradiva/MUNUS2/MUNUS2_138RacunalniskoOblikovanje_Slovarcek.pdf. [Dostopano: 9. 8. 2017].
- [35] naša lekarna. Vesna Hojnik - Morska bolezen ali potovalna slabost – ko se podatki v glavi ne ujemajo [Online]. Dosegljivo: <http://www.nasa-lekarna.si/clanki/clanek/morska-bolezen-ali-potovalna-slabost-ko-se-podatki-v-glavi-ne-ujemajo/>. [Dostopano: 3. 7. 2017].
- [36] UploadVR. Chris Reed - 7 Ways To Overcome VR Motion Sickness[Online]. Dosegljivo: <https://uploadvr.com/7-ways-overcome-vr-motion-sickness/>. [Dostopano: 31. 7. 2017].
- [37] Motion Sickness Guru. Ginger for motion sickness[Online]. Dosegljivo: <http://www.motion-sickness-guru.com/ginger-for-motion-sickness.htm>. [Dostopano: 3. 7. 2017].
- [38] MakeUseOf. Kayla Matthews - Virtual Reality Still Has 5 Big Problems To Overcome[Online]. Dosegljivo: <http://www.makeuseof.com/tag/virtual-reality-still-5-big-problems-overcome/>. [Dostopano: 31. 7. 2017].
- [39] Google Play. Eli Escobedo - VR Workout Revolution[Online]. Dosegljivo: <https://play.google.com/store/apps/details?id=com.EliJoeCompany.VRWorkoutRevolution>. [Dostopano: 26. 7. 2017].
- [40] Google Play. MyFitVR - MyFit VR[Online]. Dosegljivo: <https://play.google.com/store/apps/details?id=com.my.fit.vr.fitness>. [Dostopano: 26. 7. 2017].
- [41] Google Play. American Dream 95 - MyFit VR[Online]. Dosegljivo: <https://play.google.com/store/apps/details?id=com>.

`gymfitnessworkout2016v1995.gymfitnessworkout`. [Dostopano: 26. 7. 2017].